



INVENTORS: Owen Anderson, Craig F. Everhart, Boaz Shmueli

METHOD, APPARATUS, AND PROGRAM FOR SEPARATE  
REPRESENTATIONS OF FILE SYSTEM LOCATIONS FROM REFERRING  
FILE SYSTEMS

RELATED APPLICATIONS

5 This application is related to commonly assigned and co-pending U.S. Patent Application Serial No. 09/969,294 (Attorney Docket No. RSW920010139US1) entitled "Apparatus and Method for Offloading Application Components to Edge Servers", filed on September 28, 2001; U.S. Patent Application Serial No. 09/960,451 (Attorney Docket No. RSW920010141US1) entitled "Method and Apparatus for Minimizing Inconsistency Between Data Sources in a Web Content Distribution System", filed on September 21, 2001; and U.S. Patent Application Serial No. 09/960,448

10

(Attorney Docket No. RSW920010142US1) entitled "Method and Apparatus for Caching Subscribed and Non-Subscribed Content in a Network Data Processing System", filed on September 21, 2001, which are hereby incorporated by reference.

5

#### BACKGROUND OF THE INVENTION

##### 1. Technical Field:

The present invention relates to network file systems and, in particular, to supporting a uniform name space and transparent migration and replication of individual file systems. Still more particularly, the present invention provides a method, apparatus, and program for separate representations of file system locations from the data in referring file systems.

10  
15  
20  
25

##### 2. Description of Related Art:

A network file system is a mechanism, an architecture, that allows a computer to use files on a separate machine as if they were local to the first machine. Network file systems include a high-level network protocol that provides the structure and language for file requests between clients and servers. The protocol provides the commands for opening, reading, writing and closing files across the network and may also provide access to directory services. In network file systems that support referrals, a client may ask a server

for information about a name that appears in a first file system, as seen from the client, but is a reference to a second file system. As such, the response from the first file system must include information about the second file system, such as its location in the network. The client will then "mount" the second file system.

Mounting simply means setting up the client's operating system to do input/output (I/O) operations on a file system.

Figures 1A-1C depict example network file systems. Particularly, with reference to Figure 1A, file system 106 includes a directory "usr." The "usr" directory includes a reference to file system "foo." The reference redirects a client to file system 116.

It is assumed that objects are arranged in a tree-like structure, where files are arranged in directories and directories can contain other directories. Access to objects is achieved using path names, where a component of the path name designates the next sub-directory in the tree. The path starts at the top of the tree. A common convention uses forward slashes or back slashes to separate sub directories, and a single slash or backslash at the beginning of the path refers to the top of the hierarchy. For example, the path /a/b/C refers to an object "C" that is in directory "b." Directory "b" is in directory "a," which belongs at the top level of the hierarchy.

A file system may be moved from one location to another, such as to a new server. The referenced location must then redirect the client to the new location of the file system, and the source of the reference must be changed to indicate the new location of the file system as well. With reference to **Figure 1B**, file system 106 includes a directory "usr." The "usr" directory includes a reference to file system "foo." The reference redirects a client to file system 116. However, "foo" has moved to file system 126. Therefore, file system 116 must include information to redirect the client to file system 126. At the same time, the information in file system 106 that had directed clients to file system 116 must be changed to direct them to file system 126.

A file system may also be replicated for increased reliability. Multiple file systems can thus reference a mounted file system. For example, in **Figure 1C**, file system 106 includes a reference to file system "foo" in file system 116, file system 136 includes a reference to file system "foo" in file system 146, and file system 156 includes a reference to file system "foo" in file system 166. The file system "foo" moved to file system 126. Therefore, file systems 116, 146, 166 must be updated to include information to redirect the client to file system 126. However, too many updates results in inefficiency and a higher likelihood for error. For example, in

Figure 1C, file system 166 is not updated and the client is not redirected to the correct file system.

Similarly in Figure 1C, the references to file system "foo" in file systems 106, 136, and 156 need to be updated so as to direct clients straight to file system 126 rather than any of 116, 146, or 166.

Figures 2A and 2B illustrate examples of file systems that allow mounting on remote machines. Figure 2A depicts an example operation using Network File System (NFS), version 4 (NFSv4), which is an example of a network file system that allows mounting on remote machines. Client 208 requests an object "X" from file system (FS) server #1 206 (step 1). However, X is a mounted file system existing on FS server #2 216 and FS server #1 206 sends a redirection message identifying FS server #2 and the path, "/a/b/c/X," used to find X (step 2). Next, client 208 uses the information in the redirection message to access /a/b/c/X on server #2 (step 3).

NFSv4 requires that each referencing server include knowledge of the location and path for each mounted file system in the references returned to its clients. A server can send a redirection message that redirects the client to the server itself. This may be useful, for example, when a file system object is moved within a server. In addition, a chain of redirection messages may be used, for example, when an object is moved more than once. Thus, using NFSv4 or similar network file systems,

particularly with multiple referencing servers, the likelihood for error exists.

As another example, **Figure 2B** depicts an example operation of using the DCE's Distributed File System (DCE/DFS), which is another example of a network file system that allows referrals to remote machines. Using DCE/DFS, client 208 requests an object "X" from FS server #1 206 (step 1). However, X is a mounted file system existing on FS server #2 216 and FS server #1 206 sends an indirection response including file system identifier "Y" used to find the file system (step 2). Next, client 208 requests the location of "Y" from file system (FS) location database 220 (step 3). The FS location database returns the location of Y, "FS server #2," to client 208 (step 4). Thereafter, client 208 uses the location of FS server #2 to request the object from FS server #2 216 (step 5).

NFSv4 and similar network file systems require that a referring server know the correct locations where to direct clients. The obvious implementation of referrals in NFSv4 and similar network file systems is to embed the locations of the referenced file systems directly in the data stored in the referencing file system. The combination of the movability of the referenced file systems and the replicability of the referencing file systems makes this a cumbersome solution: if a referencing file system is replicated to many read-only locations and a referenced file system is subsequently

5 moved, all the instances of the referencing file systems  
10 must be updated even though they are read-only. DCE/DFS  
avoids this complication by storing only an identifier  
for the target file system in the referencing file  
system, so that the client looks up the current location  
for the file system given the file system identifier from  
the referencing server. It would be much less cumbersome  
for the client, not to mention conformant with NFSv4 and  
similar network file systems, if the server could handle  
the changing of file system locations without explicit  
updating of all references.

SUMMARY OF THE INVENTION

The present invention provides a file system that allows referencing between file systems which interacts well with motion and replication of file systems. A first file system includes a data object that references a second file system. The data object acts as a place holder for the second file system, but does not contain changeable data such as the name of the server containing system. The data object can be a new or existing file type with data identifying the second file system or some of its properties. The data required to locate the second file system, such as the name of a file server, is stored in a file system location data structure that is located outside the first file system. The data object then contains a key value, such as a name or a number, identifying the second file system, that can be used to look up the file system location. This allows the data in the second file system to be replicated or moved without requiring updates to the data in any redirecting or referencing servers.

The present invention also uses a basic notification mechanism to enable clients to access a uniform name space that can include all file system objects available on participating file servers. When a referencing server sends a redirection message to a client, the redirection typically includes a server location and a path. In the present invention, the referencing server encodes the

file system identification and includes the encoded file system identification rather than the path. When a server receives a request with a path that is encoded, the server decodes the file system identification. Then, 5 the server may locate the root of the file system identified by the file system identification and return the root object to the client. Location of the root can be done either by accessing the file system location data structure or by using another data structure.

10 To guarantee that clients will enter the name space at the same point, and thus view the same name space regardless of the initial participating server contacted, the present invention includes a special referral object, called the root referral object, and a special root file system. The root referral object is the top level object in all participating file servers. It contains a referral to a special designated file system 15 identification, which is the special root file system. Whenever a client contacts a participating server and 20 asks for the root object, the server will send a redirection message to the client containing the file system identification of the root file system. Since all participating file systems contain the same root file system identification, all clients will view the same name space regardless of which file server is initially 25 contacted.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

**Figures 1A-1C** depict example network file systems;

**Figures 2A** and **2B** illustrate examples of file systems that allow mounting on remote machines;

**Figure 3** depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented;

**Figure 4** is a block diagram of a data processing system that may be implemented as a server in accordance with a preferred embodiment of the present invention;

**Figure 5** is a block diagram illustrating a data processing system in which the present invention may be implemented;

**Figures 6A-6D** illustrate example file systems that allow mounting on remote machines in accordance with a preferred embodiment of the present invention; and

**Figure 7** is a flowchart illustrating the operation of a file system server in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference again to the figures, Figure 3 depicts a pictorial representation of a network of data processing systems in which the present invention may be implemented.

5 Network data processing system 300 is a network of computers in which the present invention may be implemented. Network data processing system 300 contains a network 302, which is the medium used to provide communications links between various devices and computers connected together within network data processing system 10 300. Network 302 may include connections, such as wire, wireless communication links, or fiber optic cables.

15 In the depicted example, servers 304, 314, 324 are connected to network 302. Servers 304, 314, 324 serve requests for storage units 306, 316, 326, respectively.

20 In addition, clients 308, 310, 312 are connected to network 302. These clients 308, 310, 312 may be, for example, personal computers or network computers. In the depicted example, servers 304, 314, 316 provides data stored in storage units 306, 316, 326 to clients 308-312.

25 Clients 308, 310, 312 are clients to server 304, for example. Network data processing system 300 may include additional servers, clients, and other devices not shown.

25 In the depicted example, network data processing system 300 is the Internet with network 302 representing a worldwide collection of networks and gateways that use the TCP/IP suite of protocols to communicate with one another.

At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, consisting of thousands of commercial, government, educational and other computer systems that route data and messages. Of course, network data processing system 300 also may be implemented as a number of different types of networks, such as for example, an intranet, a local area network (LAN), or a wide area network (WAN). Figure 3 is intended as an example, and not as an architectural limitation for the present invention.

Referring to Figure 4, a block diagram of a data processing system that may be implemented as a server, such as one of servers 304, 314, 324 in Figure 3, is depicted in accordance with a preferred embodiment of the present invention. Data processing system 400 may be a symmetric multiprocessor (SMP) system including a plurality of processors 402 and 404 connected to system bus 406. Alternatively, a single processor system may be employed. Also connected to system bus 406 is memory controller/ cache 408, which provides an interface to local memory 409. I/O bus bridge 410 is connected to system bus 406 and provides an interface to I/O bus 412. Memory controller/cache 408 and I/O bus bridge 410 may be integrated as depicted.

Peripheral component interconnect (PCI) bus bridge 414 connected to I/O bus 412 provides an interface to PCI local bus 416. A number of modems may be connected to PCI

local bus 416. Typical PCI bus implementations will support four PCI expansion slots or add-in connectors. Communications links to network computers 308-312 in Figure 3 may be provided through modem 418 and network adapter 420 connected to PCI local bus 416 through add-in boards.

Additional PCI bus bridges 422 and 424 provide interfaces for additional PCI local buses 426 and 428, from which additional modems or network adapters may be supported. In this manner, data processing system 400 allows connections to multiple network computers. A memory-mapped graphics adapter 430 and hard disk 432 may also be connected to I/O bus 412 as depicted, either directly or indirectly.

Those of ordinary skill in the art will appreciate that the hardware depicted in Figure 4 may vary. For example, other peripheral devices, such as optical disk drives and the like, also may be used in addition to or in place of the hardware depicted. The depicted example is not meant to imply architectural limitations with respect to the present invention.

The data processing system depicted in Figure 4 may be, for example, an IBM e-Server pSeries system, a product of International Business Machines Corporation in Armonk, New York, running the Advanced Interactive Executive (AIX) operating system or LINUX operating system.

With reference now to **Figure 5**, a block diagram illustrating a data processing system is depicted in which the present invention may be implemented. Data processing system 500 is an example of a client computer. Data processing system 500 employs a peripheral component interconnect (PCI) local bus architecture. Although the depicted example employs a PCI bus, other bus architectures such as Accelerated Graphics Port (AGP) and Industry Standard Architecture (ISA) may be used.

Processor 502 and main memory 504 are connected to PCI local bus 506 through PCI bridge 508. PCI bridge 508 also may include an integrated memory controller and cache memory for processor 502. Additional connections to PCI local bus 506 may be made through direct component interconnection or through add-in boards. In the depicted example, local area network (LAN) adapter 510, SCSI host bus adapter 512, and expansion bus interface 514 are connected to PCI local bus 506 by direct component connection. In contrast, audio adapter 516, graphics adapter 518, and audio/video adapter 519 are connected to PCI local bus 506 by add-in boards inserted into expansion slots. Expansion bus interface 514 provides a connection for a keyboard and mouse adapter 520, modem 522, and additional memory 524. Small computer system interface (SCSI) host bus adapter 512 provides a connection for hard disk drive 526, tape drive 528, and CD-ROM drive 530. Typical PCI local bus implementations will support three or four PCI expansion slots or add-in connectors.

5 An operating system runs on processor 502 and is used to coordinate and provide control of various components within data processing system 400 in **Figure 4**. The operating system may be a commercially available operating system, such as Windows 2000, which is available from Microsoft Corporation. An object oriented programming system such as Java may run in conjunction with the operating system and provide calls to the operating system from Java programs or applications executing on data processing system 500. "Java" is a trademark of Sun Microsystems, Inc. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on storage devices, such as hard disk drive 526, and may be loaded into main memory 504 for execution by processor 502.

10

15

20 Those of ordinary skill in the art will appreciate that the hardware in **Figure 5** may vary depending on the implementation. Other internal hardware or peripheral devices, such as flash ROM (or equivalent nonvolatile memory) or optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 5**. Also, the processes of the present invention may be applied to a multiprocessor data processing system.

25 As another example, data processing system 500 may be a stand-alone system configured to be bootable without relying on some type of network communication interface, whether or not data processing system 500 comprises some

type of network communication interface. As a further example, data processing system 500 may be a Personal Digital Assistant (PDA) device, which is configured with ROM and/or flash ROM in order to provide non-volatile memory for storing operating system files and/or user-generated data.

The depicted example in **Figure 5** and above-described examples are not meant to imply architectural limitations. For example, data processing system 500 also may be a notebook computer or hand held computer in addition to taking the form of a PDA. Data processing system 500 also may be a kiosk or a Web appliance.

Returning to **Figure 3**, server 304 provides access to storage 306. Similarly, server 314 provides access to storage 316 and server 324 provides access to storage 325. Storage 306 may store a first file system that includes a reference to a second file system stored in storage 316. In accordance with a preferred embodiment of the present invention, a reference object is placed in the first file system to serve as a place holder for the second file system. The reference object may be a new or existing file type with data indicating the second file system or some of its properties. This allows the normal directory enumeration procedures of the first file system to include a special name in a natural fashion.

With reference to **Figures 6A-6D**, example file systems are shown that allow mounting on remote machines in accordance with a preferred embodiment of the present

invention. Particularly, with respect to **Figure 6A**, an example is shown in which a referencing server requests location information for a referenced file system from a file system (FS) location database. Client 608 requests object "X" from FS server #1 606 (step 1). However, X is a mounted file system existing on FS server #2 616.

A referencing object may be a special file in a file system that contains an identification of a specific file system, also known as a file system identification (FSID). The FSID may be the key used to query the FS location database. The FS location database may contain, for each file system, the location of the server on which the file system resides. It may also contain the path name of the root of the file system on each server.

In the present invention, the purpose of the referencing object is to serve as a link to the root of another file system. From the client's perspective, the root of the file system that is referenced replaces the referencing object in the directory containing the referencing object. The mounted file system is thus accessed using the name of the referencing object. The client cannot itself access the referencing object using conventional file system operations. The access is manipulated by the referencing server.

FS server #1 606 sends a request for the location of X to FS location database 620 (step 2). It is convenient for the object that is in the file system on FS server #1 not to contain, itself, the information needed to the

5 redirect the client. For example, a referencing object "X" on FS server #1 may contain a key value, such as a name or number, identifying the referenced file system. FS server #1 606 may then use the key value to query FS location database 620 for the location of the file system.

10 FS location database 620 then returns the location of FS server #2 and the path "/a/b/c/X" to server 606 (step 3). Then, FS server 606 returns the location of FS server #2 and the path "/a/b/c/X" to client 608 (step 4). Client 608 then may send a request for "/a/b/c/X" to FS server #2 616 (step 5). FS server #2 may then access the file system and return the file system root to client 608. Alternatively, the location information may be found in some other data structure, such as a table in 15 memory, rather than a database.

20 The key value associated with the file system may be static. Thus, the mounted file system may be moved, put off-line, replicated, copied, or cloned, and the referencing object need not be changed. Yet, the database or table in which the attributes for the 25 referenced file system are maintained may be updated and changed, corresponding to the updates and changes of the referenced file system itself, without invalidating the data in the referencing file system.

25 In accordance with a preferred embodiment of the present invention, a basic notification mechanism is used to enable clients to access a uniform name space, which

can include all file system objects available on all participating file servers. Thus, if a client uses a certain path name to reach a file system object using any of the file servers, any other client can use the same path name to reach the same object.

Turning to **Figure 6B**, client 608 requests object "X" from FS server #1 606 (step 1). However, X is a mounted file system existing on FS server #2 616. FS server #1 606 sends a request for the location of X to FS location database 620 (step 2). FS location database 620 then returns the location of FS server #2 to server 606 (step 3).

In accordance with a preferred embodiment of the present invention, FS server #1 does not send a real path name to the client. Rather, FS server #1 606 encodes the FSID, in this case "X," using a predetermined, system wide encoding algorithm. The encoded FSID resembles a path name and can easily be decoded back to the FSID.

For example, if the FSID is a text string, such as "user.bob," a simple encoding might be to add the string "###" at the beginning and at the end of the FSID, so the path would be "###user.bob###." As such, in the example shown in **Figure 6B**, the encoded FSID would be "###X###." A requirement that disallows real path names from resembling encoded FSIDs must be enforced. A person of ordinary skill in the art will recognize that other encoding methods may also be used.

5           In **Figure 6B**, FS server 606 returns the location of FS server #2 and the path "###X###" to client 608 (step 4). The encoded FSID is transparent to the client. The client, upon receiving the name of the new server and the encoded FSID, contacts the new server and retries to access the object by passing the new server the path name given by the old server, which is the encoded FSID.

10           Thus, client 608 then sends a request for "###X###" to FS server #2 616 (step 5). FS server #2 recognizes the path name as an encoded FSID and requests the location of X from FS location database 620 (step 6). In return, FS location database 620 returns the path "/a/b/c/X" to FS server #2 616 (step 7). FS server #2 may then access the file system and return the file system root to client 608.

15           In an alternative and preferred embodiment, the location of the root of each file system can be stored in a local table that maps the FSID of each file system to a local path name. With reference to **Figure 6C**, client 608 requests object "X" from FS server #1 606 (step 1).

20           However, X is a mounted file system existing on FS server #2 616. FS server #1 606 sends a request for the location of X to FS location database 620 (step 2). FS location database 620 then returns the location of FS server #2 to server 606 (step 3).

25           FS server 606 returns the location of FS server #2 and the path "###X###" to client 608 (step 4). Client 608 then sends a request for "###X###" to FS server #2

616 (step 5). FS server #2 recognizes the path name as  
an encoded FSID and requests the location of X from FSID  
table 630 and returns the path "/a/b/c/X" to FS server #2  
616. FS server #2 may then access the file system and  
return the file system root to client 608.

5  
10  
15  
20  
25  
30  
35  
40  
45  
50  
55  
60  
65  
70  
75  
80  
85  
90  
95  
100  
105  
110  
115  
120  
125  
130  
135  
140  
145  
150  
155  
160  
165  
170  
175  
180  
185  
190  
195  
200  
205  
210  
215  
220  
225  
230  
235  
240  
245  
250  
255  
260  
265  
270  
275  
280  
285  
290  
295  
300  
305  
310  
315  
320  
325  
330  
335  
340  
345  
350  
355  
360  
365  
370  
375  
380  
385  
390  
395  
400  
405  
410  
415  
420  
425  
430  
435  
440  
445  
450  
455  
460  
465  
470  
475  
480  
485  
490  
495  
500  
505  
510  
515  
520  
525  
530  
535  
540  
545  
550  
555  
560  
565  
570  
575  
580  
585  
590  
595  
600  
605  
610  
615  
620  
625  
630  
635  
640  
645  
650  
655  
660  
665  
670  
675  
680  
685  
690  
695  
700  
705  
710  
715  
720  
725  
730  
735  
740  
745  
750  
755  
760  
765  
770  
775  
780  
785  
790  
795  
800  
805  
810  
815  
820  
825  
830  
835  
840  
845  
850  
855  
860  
865  
870  
875  
880  
885  
890  
895  
900  
905  
910  
915  
920  
925  
930  
935  
940  
945  
950  
955  
960  
965  
970  
975  
980  
985  
990  
995  
1000  
1005  
1010  
1015  
1020  
1025  
1030  
1035  
1040  
1045  
1050  
1055  
1060  
1065  
1070  
1075  
1080  
1085  
1090  
1095  
1100  
1105  
1110  
1115  
1120  
1125  
1130  
1135  
1140  
1145  
1150  
1155  
1160  
1165  
1170  
1175  
1180  
1185  
1190  
1195  
1200  
1205  
1210  
1215  
1220  
1225  
1230  
1235  
1240  
1245  
1250  
1255  
1260  
1265  
1270  
1275  
1280  
1285  
1290  
1295  
1300  
1305  
1310  
1315  
1320  
1325  
1330  
1335  
1340  
1345  
1350  
1355  
1360  
1365  
1370  
1375  
1380  
1385  
1390  
1395  
1400  
1405  
1410  
1415  
1420  
1425  
1430  
1435  
1440  
1445  
1450  
1455  
1460  
1465  
1470  
1475  
1480  
1485  
1490  
1495  
1500  
1505  
1510  
1515  
1520  
1525  
1530  
1535  
1540  
1545  
1550  
1555  
1560  
1565  
1570  
1575  
1580  
1585  
1590  
1595  
1600  
1605  
1610  
1615  
1620  
1625  
1630  
1635  
1640  
1645  
1650  
1655  
1660  
1665  
1670  
1675  
1680  
1685  
1690  
1695  
1700  
1705  
1710  
1715  
1720  
1725  
1730  
1735  
1740  
1745  
1750  
1755  
1760  
1765  
1770  
1775  
1780  
1785  
1790  
1795  
1800  
1805  
1810  
1815  
1820  
1825  
1830  
1835  
1840  
1845  
1850  
1855  
1860  
1865  
1870  
1875  
1880  
1885  
1890  
1895  
1900  
1905  
1910  
1915  
1920  
1925  
1930  
1935  
1940  
1945  
1950  
1955  
1960  
1965  
1970  
1975  
1980  
1985  
1990  
1995  
2000  
2005  
2010  
2015  
2020  
2025  
2030  
2035  
2040  
2045  
2050  
2055  
2060  
2065  
2070  
2075  
2080  
2085  
2090  
2095  
2100  
2105  
2110  
2115  
2120  
2125  
2130  
2135  
2140  
2145  
2150  
2155  
2160  
2165  
2170  
2175  
2180  
2185  
2190  
2195  
2200  
2205  
2210  
2215  
2220  
2225  
2230  
2235  
2240  
2245  
2250  
2255  
2260  
2265  
2270  
2275  
2280  
2285  
2290  
2295  
2300  
2305  
2310  
2315  
2320  
2325  
2330  
2335  
2340  
2345  
2350  
2355  
2360  
2365  
2370  
2375  
2380  
2385  
2390  
2395  
2400  
2405  
2410  
2415  
2420  
2425  
2430  
2435  
2440  
2445  
2450  
2455  
2460  
2465  
2470  
2475  
2480  
2485  
2490  
2495  
2500  
2505  
2510  
2515  
2520  
2525  
2530  
2535  
2540  
2545  
2550  
2555  
2560  
2565  
2570  
2575  
2580  
2585  
2590  
2595  
2600  
2605  
2610  
2615  
2620  
2625  
2630  
2635  
2640  
2645  
2650  
2655  
2660  
2665  
2670  
2675  
2680  
2685  
2690  
2695  
2700  
2705  
2710  
2715  
2720  
2725  
2730  
2735  
2740  
2745  
2750  
2755  
2760  
2765  
2770  
2775  
2780  
2785  
2790  
2795  
2800  
2805  
2810  
2815  
2820  
2825  
2830  
2835  
2840  
2845  
2850  
2855  
2860  
2865  
2870  
2875  
2880  
2885  
2890  
2895  
2900  
2905  
2910  
2915  
2920  
2925  
2930  
2935  
2940  
2945  
2950  
2955  
2960  
2965  
2970  
2975  
2980  
2985  
2990  
2995  
3000  
3005  
3010  
3015  
3020  
3025  
3030  
3035  
3040  
3045  
3050  
3055  
3060  
3065  
3070  
3075  
3080  
3085  
3090  
3095  
3100  
3105  
3110  
3115  
3120  
3125  
3130  
3135  
3140  
3145  
3150  
3155  
3160  
3165  
3170  
3175  
3180  
3185  
3190  
3195  
3200  
3205  
3210  
3215  
3220  
3225  
3230  
3235  
3240  
3245  
3250  
3255  
3260  
3265  
3270  
3275  
3280  
3285  
3290  
3295  
3300  
3305  
3310  
3315  
3320  
3325  
3330  
3335  
3340  
3345  
3350  
3355  
3360  
3365  
3370  
3375  
3380  
3385  
3390  
3395  
3400  
3405  
3410  
3415  
3420  
3425  
3430  
3435  
3440  
3445  
3450  
3455  
3460  
3465  
3470  
3475  
3480  
3485  
3490  
3495  
3500  
3505  
3510  
3515  
3520  
3525  
3530  
3535  
3540  
3545  
3550  
3555  
3560  
3565  
3570  
3575  
3580  
3585  
3590  
3595  
3600  
3605  
3610  
3615  
3620  
3625  
3630  
3635  
3640  
3645  
3650  
3655  
3660  
3665  
3670  
3675  
3680  
3685  
3690  
3695  
3700  
3705  
3710  
3715  
3720  
3725  
3730  
3735  
3740  
3745  
3750  
3755  
3760  
3765  
3770  
3775  
3780  
3785  
3790  
3795  
3800  
3805  
3810  
3815  
3820  
3825  
3830  
3835  
3840  
3845  
3850  
3855  
3860  
3865  
3870  
3875  
3880  
3885  
3890  
3895  
3900  
3905  
3910  
3915  
3920  
3925  
3930  
3935  
3940  
3945  
3950  
3955  
3960  
3965  
3970  
3975  
3980  
3985  
3990  
3995  
4000  
4005  
4010  
4015  
4020  
4025  
4030  
4035  
4040  
4045  
4050  
4055  
4060  
4065  
4070  
4075  
4080  
4085  
4090  
4095  
4100  
4105  
4110  
4115  
4120  
4125  
4130  
4135  
4140  
4145  
4150  
4155  
4160  
4165  
4170  
4175  
4180  
4185  
4190  
4195  
4200  
4205  
4210  
4215  
4220  
4225  
4230  
4235  
4240  
4245  
4250  
4255  
4260  
4265  
4270  
4275  
4280  
4285  
4290  
4295  
4300  
4305  
4310  
4315  
4320  
4325  
4330  
4335  
4340  
4345  
4350  
4355  
4360  
4365  
4370  
4375  
4380  
4385  
4390  
4395  
4400  
4405  
4410  
4415  
4420  
4425  
4430  
4435  
4440  
4445  
4450  
4455  
4460  
4465  
4470  
4475  
4480  
4485  
4490  
4495  
4500  
4505  
4510  
4515  
4520  
4525  
4530  
4535  
4540  
4545  
4550  
4555  
4560  
4565  
4570  
4575  
4580  
4585  
4590  
4595  
4600  
4605  
4610  
4615  
4620  
4625  
4630  
4635  
4640  
4645  
4650  
4655  
4660  
4665  
4670  
4675  
4680  
4685  
4690  
4695  
4700  
4705  
4710  
4715  
4720  
4725  
4730  
4735  
4740  
4745  
4750  
4755  
4760  
4765  
4770  
4775  
4780  
4785  
4790  
4795  
4800  
4805  
4810  
4815  
4820  
4825  
4830  
4835  
4840  
4845  
4850  
4855  
4860  
4865  
4870  
4875  
4880  
4885  
4890  
4895  
4900  
4905  
4910  
4915  
4920  
4925  
4930  
4935  
4940  
4945  
4950  
4955  
4960  
4965  
4970  
4975  
4980  
4985  
4990  
4995  
5000  
5005  
5010  
5015  
5020  
5025  
5030  
5035  
5040  
5045  
5050  
5055  
5060  
5065  
5070  
5075  
5080  
5085  
5090  
5095  
5100  
5105  
5110  
5115  
5120  
5125  
5130  
5135  
5140  
5145  
5150  
5155  
5160  
5165  
5170  
5175  
5180  
5185  
5190  
5195  
5200  
5205  
5210  
5215  
5220  
5225  
5230  
5235  
5240  
5245  
5250  
5255  
5260  
5265  
5270  
5275  
5280  
5285  
5290  
5295  
5300  
5305  
5310  
5315  
5320  
5325  
5330  
5335  
5340  
5345  
5350  
5355  
5360  
5365  
5370  
5375  
5380  
5385  
5390  
5395  
5400  
5405  
5410  
5415  
5420  
5425  
5430  
5435  
5440  
5445  
5450  
5455  
5460  
5465  
5470  
5475  
5480  
5485  
5490  
5495  
5500  
5505  
5510  
5515  
5520  
5525  
5530  
5535  
5540  
5545  
5550  
5555  
5560  
5565  
5570  
5575  
5580  
5585  
5590  
5595  
5600  
5605  
5610  
5615  
5620  
5625  
5630  
5635  
5640  
5645  
5650  
5655  
5660  
5665  
5670  
5675  
5680  
5685  
5690  
5695  
5700  
5705  
5710  
5715  
5720  
5725  
5730  
5735  
5740  
5745  
5750  
5755  
5760  
5765  
5770  
5775  
5780  
5785  
5790  
5795  
5800  
5805  
5810  
5815  
5820  
5825  
5830  
5835  
5840  
5845  
5850  
5855  
5860  
5865  
5870  
5875  
5880  
5885  
5890  
5895  
5900  
5905  
5910  
5915  
5920  
5925  
5930  
5935  
5940  
5945  
5950  
5955  
5960  
5965  
5970  
5975  
5980  
5985  
5990  
5995  
6000  
6005  
6010  
6015  
6020  
6025  
6030  
6035  
6040  
6045  
6050  
6055  
6060  
6065  
6070  
6075  
6080  
6085  
6090  
6095  
6100  
6105  
6110  
6115  
6120  
6125  
6130  
6135  
6140  
6145  
6150  
6155  
6160  
6165  
6170  
6175  
6180  
6185  
6190  
6195  
6200  
6205  
6210  
6215  
6220  
6225  
6230  
6235  
6240  
6245  
6250  
6255  
6260  
6265  
6270  
6275  
6280  
6285  
6290  
6295  
6300  
6305  
6310  
6315  
6320  
6325  
6330  
6335  
6340  
6345  
6350  
6355  
6360  
6365  
6370  
6375  
6380  
6385  
6390  
6395  
6400  
6405  
6410  
6415  
6420  
6425  
6430  
6435  
6440  
6445  
6450  
6455  
6460  
6465  
6470  
6475  
6480  
6485  
6490  
6495  
6500  
6505  
6510  
6515  
6520  
6525  
6530  
6535  
6540  
6545  
6550  
6555  
6560  
6565  
6570  
6575  
6580  
6585  
6590  
6595  
6600  
6605  
6610  
6615  
6620  
6625  
6630  
6635  
6640  
6645  
6650  
6655  
6660  
6665  
6670  
6675  
6680  
6685  
6690  
6695  
6700  
6705  
6710  
6715  
6720  
6725  
6730  
6735  
6740  
6745  
6750  
6755  
6760  
6765  
6770  
6775  
6780  
6785  
6790  
6795  
6800  
6805  
6810  
6815  
6820  
6825  
6830  
6835  
6840  
6845  
6850  
6855  
6860  
6865  
6870  
6875  
6880  
6885  
6890  
6895  
6900  
6905  
6910  
6915  
6920  
6925  
6930  
6935  
6940  
6945  
6950  
6955  
6960  
6965  
6970  
6975  
6980  
6985  
6990  
6995  
7000  
7005  
7010  
7015  
7020  
7025  
7030  
7035  
7040  
7045  
7050  
7055  
7060  
7065  
7070  
7075  
7080  
7085  
7090  
7095  
7100  
7105  
7110  
7115  
7120  
7125  
7130  
7135  
7140  
7145  
7150  
7155  
7160  
7165  
7170  
7175  
7180  
7185  
7190  
7195  
7200  
7205  
7210  
7215  
7220  
7225  
7230  
7235  
7240  
7245  
7250  
7255  
7260  
7265  
7270  
7275  
7280  
7285  
7290  
7295  
7300  
7305  
7310  
7315  
7320  
7325  
7330  
7335  
7340  
7345  
7350  
7355  
7360  
7365  
7370  
7375  
7380  
7385  
7390  
7395  
7400  
7405  
7410  
7415  
7420  
7425  
7430  
7435  
7440  
7445  
7450  
7455  
7460  
7465  
7470  
7475  
7480  
7485  
7490  
7495  
7500  
7505  
7510  
7515  
7520  
7525  
7530  
7535  
7540  
7545  
7550  
7555  
7560  
7565  
7570  
7575  
7580  
7585  
7590  
7595  
7600  
7605  
7610  
7615  
7620  
7625  
7630  
7635  
7640  
7645  
7650  
7655  
7660  
7665  
7670  
7675  
7680  
7685  
7690  
7695  
7700  
7705  
7710  
7715  
7720  
7725  
7730  
7735  
7740  
7745  
7750  
7755  
7760  
7765  
7770  
7775  
7780  
7785  
7790  
7795  
7800  
7805  
7810  
7815  
7820  
7825  
7830  
7835  
7840  
7845  
7850  
7855  
7860  
7865  
7870  
7875  
7880  
7885  
7890  
7895  
7900  
7905  
7910  
7915  
7920  
7925  
7930  
7935  
7940  
7945  
7950  
7955  
7960  
7965  
7970  
7975  
7980  
7985  
7990  
7995  
8000  
8005  
8010  
8015  
8020  
8025  
8030  
8035  
8040  
8045  
8050  
8055  
8060  
8065  
8070  
8075  
8080  
8085  
8090  
8095  
8100  
8105  
8110  
8115  
8120  
8125  
8130  
8135  
8140  
8145  
8150  
8155  
8160  
8165  
8170  
8175  
8180  
8185  
8190  
8195  
8200  
8205  
8210  
8215  
8220  
8225  
8230  
8235  
8240  
8245  
8250  
8255  
8260  
8265  
8270  
8275  
8280  
8285  
8290  
8295  
8300  
8305  
8310  
8315  
8320  
8325  
8330  
8335  
8340  
8345  
8350  
8355  
8360  
8365  
8370  
8375  
8380  
8385  
8390  
8395  
8400  
8405  
8410  
8415  
8420  
8425  
8430  
8435  
8440  
8445  
8450  
8455  
8460  
8465  
8470  
8475  
8480  
8485  
8490  
8495  
8500  
8505  
8510  
8515  
8520  
8525  
8530  
8535  
8540  
8545  
8550  
8555  
8560  
8565  
8570  
8575  
8580  
8585  
8590  
8595  
8600  
8605  
8610  
8615  
8620  
8625  
8630  
8635  
8640  
8645  
8650  
8655  
8660  
8665  
8670  
8675  
8680  
8685  
8690  
8695  
8700  
8705  
8710  
8715  
8720  
8725  
8730  
8735  
8740  
8745  
8750  
8755  
8760  
8765  
8770  
8775  
8780  
8785  
8790  
8795  
8800  
8805  
8810  
8815  
8820  
8825  
8830  
8835  
8840  
8845  
8850  
8855  
8860  
8865  
8870  
8875  
8880  
8885  
8890  
8895  
8900  
8905  
8910  
8915  
8920  
8925  
8930  
8935  
8940  
8945  
8950  
8955  
8960  
8965  
8970  
8975  
8980  
8985  
8990  
8995  
9000  
9005  
9010  
9015  
9020  
9025  
9030  
9035  
9040  
9045  
9050  
9055  
9060  
9065  
9070  
9075  
9080  
9085  
9090  
9095  
9100  
9105  
9110  
9115  
9120  
9125  
9130  
9135  
9140  
9145  
9150  
9155  
9160  
9165  
9170  
9175  
9180  
9185  
9190  
9195  
9200  
9205  
9210  
9215  
9220  
9225  
9230  
9235  
9240  
9245  
9250  
9255  
9260  
9265  
9270  
9275  
9280  
9285  
9290  
9295  
9300  
9305  
9310  
9315  
9320  
9325  
9330  
9335  
9340  
9345  
9350  
9355  
9360  
9365  
9370  
9375  
9380  
9385  
9390  
9395  
9400  
9405  
9410  
9415  
9420  
9425  
9430  
9435  
9440  
9445  
9450  
9455  
9460  
9465  
9470  
9475  
9480  
9485  
9490  
9495  
9500  
9505  
9510  
9515  
9520  
9525  
9530  
9535  
9540  
9545  
9550  
9555

now to **Figure 6D**, client 608 requests the top level object from FS server #1 606 (step 1). FS server #1 606 accesses the root referencing object and sends a request to FS location database 620 for the location of the file system with the FSID of "rootfs" (step 2). The FS location database returns the location of rootfs, FS server #2, to FS server #1 606 (step 3). FS server #1 then encodes the FSID to form "###rootfs###" and sends a redirection message indicating that the file system is on FS server #2 and the path is "###rootfs###" to client 608 (step 4).

Next, client 608 sends a request for "###rootfs###" to FS server #2 616 (step 5). FS server #2 recognizes the path name as an encoded FSID and sends a request for the location of the rootfs file system to FS location database 620 (step 6). The FS location database then returns the path, "/a/b/c/," to FS server #2 616 (step 7). FS server #2 may then access the file system and return the file system root to client 608. In an alternative and preferred embodiment, the location of the root of the file system can be stored in a local table that maps the FSID of each file system to a local path name.

The example in **Figure 6D** shows only two file system servers. However, fewer or more participating servers may be included. For example, the network file system may include only a single server, such as server 616. If a client requests the top level object of FS server #2

616, the server may send a redirection message  
redirecting the client to itself with the encoded FSID,  
"###rootfs###." The example of **Figure 6D** may also  
include more participating file system servers, each of  
5 which has the same referencing object as its top level  
object.

With reference to **Figure 7**, a flowchart is shown  
illustrating the operation of a file system server in  
accordance with a preferred embodiment of the present  
invention. The process begins and receives a request for  
10 an object (step 702). A determination is made as to  
whether the object is a reference to another file system  
(step 704). If the object is a reference to a separate  
file system, the process looks up the location of the  
separate file system (step 706) and encodes the FSID of  
the file system (step 708). Next, the process returns a  
15 redirection message indicating the location and the  
encoded FSID as the path name to the client (step 710)  
and ends.

20 If the object is not a referencing object in step  
704, a determination is made as to whether the object is  
a top level object (step 712). If the object is a top  
level object, the process accesses the root referencing  
object indicating an FSID of "rootfs" (step 714). Then,  
25 the process looks up the location of the file system  
(step 706) and encodes the FSID of the file system (step  
708). Thereafter, the process returns a redirection

message indicating the location and the encoded FSID as the path name to the client (step 710) and ends.

5        If the object is not a top level object in step 712, a determination is made as to whether the path is an encoded FSID (step 716). If the path is an encoded FSID, the process decodes the path to form the FSID (step 718), accesses and returns the root of the file system (step 720), and ends. If the path is not an encoded FSID in step 716, the process accesses and returns the real object (step 722) and ends.

10        A referencing object may refer to a file system on an unmodified server that does not incorporate the present invention. This may be achieved by having the referencing object itself include the location of the unmodified server and the path name of the file system on that server. When a client accesses the referencing object, the server will reply with the information included in the FS location database for the file system on the unmodified server.

20        The referencing object may also include an FSID, as if it referred to a modified server. However, the file system database may contain a special tag that marks the file system stored on an unmodified server, together with the server location and path name. The server may then reply to the client with the information retrieved from the FS location database. However, the path name cannot comprise an encoded FSID, and the unmodified server may

not be configured to contain referrals to other file systems.

5 The present invention may also support file systems that are replicated on various modified servers. If the network file system protocol supports a redirection message that includes more than one server, such as is the case with NFSv4, the present invention may be extended to include the locations of all the servers that host a particular file system. When a client attempts to access a referencing object, the modified server may reply with a list of file system server locations. The client may then select a server to access.

10 15 20 Alternatively, if the network file system protocol does not support a redirection message that includes more than one server location, the modified server, after receiving the list of server locations from the FS location database, may choose which server to include in the redirection message. The selection algorithm may be, for example, a round robin algorithm, and may be based on various factors, such as server load and server response time.

25 Thus, the present invention solves the disadvantages of the prior art by providing a data object that references a second file system that is interpreted directly by the file server. The data required to locate the second file system is stored in a file system location data structure that may be located outside the first file system. The data object may then contain a

key value, such as a name or a number, identifying the second file system, that can be used to look up the file system location. Therefore, a file system may be referred to by another file system transparently to the client.

A referencing server encodes the file system identification and includes the encoded file system identification rather than a path. When a server receives a request with a path that is encoded, the server decodes the file system identification. Then, the server may locate the root of the file system identified by the file system identification and return the root object to the client. Location of the root can be done either by accessing the file system location database structure or by using another, local data structure. A root referral object is the top level object in all participating file servers. It contains a referral to a root file system identification, which is the root file system. Since all participating file servers contain the same root file system identification, all clients will view the same name space regardless of which file server is initially contacted.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes of the present invention are capable of being distributed in the form of a computer readable medium of instructions

and a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to carry out the distribution. Examples of computer readable media include recordable-type media, such as a floppy disk, a hard disk drive, a RAM, CD-ROMs, DVD-ROMs, and transmission-type media, such as digital and analog communications links, wired or wireless communications links using transmission forms, such as, for example, radio frequency and light wave transmissions. The computer readable media may take the form of coded formats that are decoded for actual use in a particular data processing system.

The description of the present invention has been presented for purposes of illustration and description, and is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for various embodiments with various modifications as are suited to the particular use contemplated.